

MANATT, PHELPS & PHILLIPS, LLP
ROBERT D. BECKER (Bar No. CA 160648)
E-mail: rbecker@manatt.com
RONALD S. KATZ (Bar No. CA 085713)
E-mail: rkatz@manatt.com
SHAWN G. HANSEN (Bar No. CA 197033)
E-mail: shansen@manatt.com
1001 Page Mill Road, Building 2
Palo Alto, CA 94304-1006
Telephone: (650) 812-1300
Facsimile: (650) 213-0260

Attorneys for
KELORA SYSTEMS, LLC

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA
OAKLAND DIVISION

_____) No. 4:10-cv-4947-CW (filed Nov. 2, 2010)
eBay Inc. and Microsoft Corporation,)
)
) **DECLARATION OF KRIS KIMBROUGH**
) **IN SUPPORT OF KELORA'S**
) **OPPOSITION TO DEFENDANTS'**
) **CLAIM CONSTRUCTION BRIEF AND**
) **MOTION FOR SUMMARY JUDGMENT**
) **OF INVALIDITY AND**
) **NONINFRINGEMENT**
)
)
)
Plaintiffs and Counterclaim-
Defendants,

v.

Kelora Systems, LLC,

Defendant and Counterclaim-Plaintiff.)

_____) No. 4:11-cv-1398-CW (filed Mar. 23, 2011)
Cabela's Inc.,)
) (related case)
)
)
) *Plaintiff and Counterclaim-Defendant,*
)
)
)
v.
)
)
)
Kelora Systems, LLC,
)
)
)
Defendant and Counterclaim-Plaintiff.)

1 Kelora Systems, LLC,) No. 4:11-cv-1548-CW (filed Nov. 8, 2010)
 2) (related case)

3 *Plaintiff and Counterclaim-Defendant,*)

4 v.)

5 Target Corporation; OfficeMax)
 6 Incorporated; Rockler Companies, Inc.; 1-)
 7 800-Flowers.com, Inc.; Amazon.com, Inc.;)
 8 Dell, Inc.; Office Depot, Inc.; Newegg Inc.;)
 9 Costco Wholesale Corporation; Hewlett-)
 10 Packard Company; CircuitCity.com Inc.;)
 11 Audible, Inc.; and Zappos.com, Inc.,)

12 *Defendants and Counterclaim-*)
 13 *Plaintiffs.*)

14 OfficeMax Incorporated,)

15 *Third-Party Plaintiff,*)

16 v.)

17 Adobe Systems Incorporated,)

18 *Third-Party Defendant.*)

19) No. 4:11-cv-2284-CW (filed Feb. 3, 2011)
 20) (related case)

21 Nebraska Furniture Mart, Inc.,)
 22 *Plaintiff and Counterclaim-Defendant,*)

23 v.)

24 Kelora Systems, LLC,)

25 *Defendant and Counterclaim-Plaintiff.*)

26 I, Kris Kimbrough, declare as follows:

27 1. I am a named inventor on the patent at issue in this case, U.S. Patent No.
 28 6,275,821 (“the ‘821 patent”). I have personal knowledge of the following facts. If called as a
 witness, I could and would testify competently to these facts.

2. I have reviewed the Declaration of Theodore Chandler (“Chandler Declaration”) submitted in support of Defendants’ Claim Construction Brief and Motion for Summary Judgment of Invalidity and Noninfringement (“Defendants’ Motion”). Mr. Chandler’s

1 Declaration relies on source code included on two CD-ROMs attached as Exhibits Q and R
2 (containing compressed and uncompressed versions).

3 3. I am the author of the source code on the CD-ROMs attached to the Chandler
4 Declaration as Exhibits Q and R.

5 4. Attached as Exhibit A to this declaration is a true and correct copy of the source
6 code files NAVIGATE.C and CPC.INC included on those CD-ROMs, with the exception of the
7 added title headers, page numbers and line numbers that are not part of the original files. Exhibit
8 A shows the source code files NAVIGATE.C and CPC.INC, written in the C programming
9 language, that can be used to compile the executable code for the AMP Navigator demonstration
10 program.

11 5. I was deposed on January 22, 2009, in conjunction with *PartsRiver, Inc., v.*
12 *Shopzilla, Inc., et al.*, Case No. 2-07-CV-440 DF in the United States District Court of the Eastern
13 District of Texas ("Original PartsRiver Litigation").

14 6. At the time of my deposition on January 22, 2009, I had not recently reviewed the
15 source code for the AMP Navigator demonstration program and due to the passage of time did
16 not remember it with certainty. In that deposition, I testified as follows without the benefit of
17 reviewing the source code files:

18 Mr. Kimbrough: I think every version we built always did
19 concatenated search.

20 Mr. Cederoth: You think. Did the -- did the 1992
21 demonstration -- the software you found, did that do concatenated
22 search?

22 A. I believe it did. I didn't review the -- the source code.

23 Q. So you think all your work always did concatenated
24 searching?

24 A. I believe so.

25 Deposition of K. Kimbrough, January 22, 2009, 118:14-23
26 (Emphasis added).

27 7. I have reviewed the source code files NAVIGATE.C and CPC.INC for the AMP
28 Navigator demonstration program subsequent to my deposition on January 22, 2009, and have

1 now confirmed that my tentative belief regarding concatenated search stated in my above-quoted
2 deposition testimony was incorrect.

3 8. It is my understanding now and it was my understanding at the time of that
4 deposition that “concatenated search” referred to two or more search terms submitted together for
5 processing. However, as explained more fully below, based on my review since the deposition of
6 the source code files NAVIGATE.C and CPC.INC for the AMP Navigator demonstration
7 program, taken from the CD-ROM attached to the Chandler Declaration, I have confirmed that
8 the AMP Navigator demonstration program, when run, does not in fact perform “concatenated
9 search” and is not capable of doing so. As explained more fully below, the source code shows
10 that once a feature has been submitted for searching, the status of the feature is changed from
11 MARKED to SELECTED for the remainder of the session and the feature is not submitted again
12 for searching. This is because a feature is qualified to be submitted for the program to use for
13 searching *only if* the feature’s record is designated as MARKED. Accordingly, previously
14 searched features, which remain designated as SELECTED, are excluded and are not submitted
15 again for searching.

16 9. In addition, as explained below more fully, I am able to confirm based on the
17 source code files NAVIGATE.C and CPC.INC for the AMP Navigator demonstration program
18 that the “resubmission” step as recited in step (h) of both claims 1 and 9 of the reexamined ‘821
19 patent is not performed and cannot be performed when the AMP Navigator demonstration
20 program is run and used. The term “resubmission” was not discussed during my deposition of
21 January 22, 2009.

22 **Independent Claims 1 and 9 of the Reexamined ‘821 Patent**

23 10. Reexamined claims 1 and 9 of the ‘821 patent specify in their preambles that they
24 are performed “with a server.” For example, claim 1 states that the claimed method is
25 “performed with a server connected to a client computer through a computer network.” Claim 9
26 states that the claimed method is “performed with a server connected to a computer network.”

27 11. Step (h) of claim 1 of the reexamined ‘821 patent recites “accepting a second
28 selection criteria from said client computer via said computer network at said server wherein the

1 second selection criteria comprises a resubmission to the server of the alternative or alternatives
2 of the first selection criteria plus at least one alternative selected from the revised feature screen.”

3 12. Step (h) of claim 9 of the reexamined ‘821 patent recites “receiving and accepting
4 a second selection criteria from said client computer via said computer network, in which said
5 second selection criteria comprises (1) a resubmission by said client computer of the alternative
6 or alternatives of the first selection criteria along with (2) at least one alternative selected from the
7 revised feature screen”.

8 13. According to its ordinary and customary meaning, the term “resubmission” as
9 recited in step (h) refers to a subsequent submission of something which was the object of a
10 previous submission.

11 **AMP Navigator Demonstration Program Source Code**

12 14. Once the source code for the AMP Navigator demonstration program is compiled
13 into object code, which is linked into executable code, which is run by a computer, the AMP
14 Navigator demonstration program does not perform any client-server interaction. The source
15 code does not have any server functionality.

16 15. The source code for the AMP Navigator demonstration program shows that radio
17 buttons and/or list boxes associated with features are sent to the software window of a user
18 interface.

19 16. The source code file CPC.INC shows the data structure for the AMP Navigator
20 demonstration program that is used for recording the selection status of all the features in the
21 software window. Exhibit A, pp. 16 – 18. Possible statuses for a feature include SELECTABLE,
22 indicating the feature is available to be selected, MARKED, indicating that the feature was
23 selected but a search has not been run on the feature, and SELECTED, indicating that a search
24 has been run on the feature in a previous request in this search session.

25 17. The source code file NAVIGATE.C for the AMP Navigator demonstration
26 program shows that as the program determines that a particular feature’s radio button or list box
27 entry in the software window is newly selected, the feature’s record in the data structure is
28 designated with the “MARKED” attribute by running the instruction:

Features[i].CurState = MARKED;

to modify each such feature's record. Exhibit A, p. 9, Line 12, and p. 11, Line 33.

18. The source code file NAVIGATE.C for the AMP Navigator demonstration program shows that after a request is received to run a first search using the features that were selected in the software window, a feature is determined to be qualified for submitting to the program for use in searching *only if* the feature's record is designated as MARKED. The program checks for qualifying features by running the instruction:

if (Features[i].CurState == MARKED)

through each feature's record to check its status. Exhibit A, p. 9, Line 27.

19. The source code file NAVIGATE.C for the AMP Navigator demonstration program shows that each qualifying feature is submitted, one at a time, for the program to use for searching against the database and to retrieve results that match the feature. The program submits one feature at a time to search against the database by running the instruction:

dXgetrno(IndexPtr[Features[i].DBFeatureNum], Features[i].Abbrev,
&RecNum);

Exhibit A, p. 9, Lines 42 – 43.

20. The source code file NAVIGATE.C for the AMP Navigator demonstration program shows that after searching for matches for a first feature, a first set of search results are returned for the feature. After searching for matches for a second feature, a second set of search results are returned for the second feature. The second set of search results are intersected with the first set of search results to generate a main set by running the instruction:

MergeRecordLists();

Exhibit A, p. 9, Line 57. Each set of subsequent search results returned for another feature is also intersected with the main set by running the instruction:

MergeRecordLists();

for each result set. Exhibit A, *id.* When one set is intersected with another set, only results found in both sets are kept. When all the result sets are intersected, the first round of searching in this search session is completed, and the software window is revised.

21. The source code file NAVIGATE.C for the AMP Navigator demonstration program shows that after a feature is used for the first search request, the status for the searched feature is changed to **SELECTED** by running the instruction:

```
Features[i].CurState = SELECTED;
```

Exhibit A, p. 10, line 1.

22. The source code file NAVIGATE.C for the AMP Navigator demonstration program shows that after a first round of searching in this search session is completed, the portion of the source code described in Paragraph 17 continues to be executed such that as the program determines a particular feature's radio button or list box entry in the software window is being selected, the feature's record in the data structure is updated and designated with the "MARKED" attribute. However, the program will *skip over* any features that had been previously searched in this search session, and will not designate any such feature with the "MARKED" attribute. For example, the program runs the instruction:

```
if (Features[i].CurState != SELECTED)
```

(i.e., if the status is not "SELECTED") at various points in the source code to exclude previously searched features from being designated with the "MARKED" attribute. *See, e.g.*, Exhibit A, p. 9, line 5, and p. 11, line 31. This instruction means that the previously searched features remain designated as "SELECTED" per Paragraph 21, and are not again designated as "MARKED" for the next round of searching.

23. As shown by the source code file NAVIGATE.C for the AMP Navigator demonstration program and described in Paragraph 18, a feature is qualified to be submitted for the program to use for searching *only if* the feature's record is designated as MARKED. Accordingly, previously searched features which remain designated as SELECTED are excluded and are not submitted again for searching.

24. As shown in the source code file NAVIGATE.C for the AMP Navigator demonstration program described in Paragraph 20, each set of search results returned for a new feature is intersected into the main set by running the instruction:

```
MergeRecordLists();
```

1 for each result set. Exhibit A, p. 9, Line 57.

2 25. No previously searched records are submitted again in the second round of
3 searching because such records are excluded by the program in the second round. Accordingly,
4 because nothing previously submitted is submitted again, no "resubmission" occurs for the
5 second round when the source code for the AMP Navigator demonstration program is run and
6 used.

7 26. Each qualifying feature is submitted, one at a time, for the program to use for
8 searching against the database and to retrieve results that match the feature. As two or more
9 search terms are never submitted together for processing, no "concatenated search" occurs when
10 the source code for the AMP Navigator demonstration program is run and used.

11 27. While the AMP Navigator demonstration program is running, the user interface
12 shows only the appearance of certain user interface elements, such as radio buttons as selected or
13 not selected and/or radio buttons made unselectable by graying them out. It is not possible to
14 determine from observation of the user interface whether "concatenated search" or
15 "resubmission" is being performed. As discussed above, when the source code for the AMP
16 Navigator demonstration program is run and used, it is not capable of performing those functions.
17 If one were to take screen shots of the user interface while the AMP Navigator demonstration
18 program is running, it would not be possible to determine whether or how search is being
19 performed.

20 I declare under penalty of perjury that the foregoing is true and correct.

21 Executed this 6 day of October 2011, at Sunnyvale, California.

22
23
24 By: 

Kris Kimbrough

25
26
27
28 300764792.3